Proceedings of the 14th International Symposium on Process Systems Engineering – PSE 2021+
June 19-23, 2022, Kyoto, Japan

Advances in Generalized Disjunctive and Mixed-Integer Nonlinear Programming Algorithms and Software for Superstructure Optimization

David E. Bernal[a], Yunshan Liu[a], Michael L. Bynum[b], Carl D. Laird[a], John D. Siirola[b], and Ignacio E. Grossmann[a]

[a]*Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213, United States of America*
[b]*Discrete Mathematics and Optimization, Sandia National Laboratories, 1515 Eubank SE, Albuquerque NM, 87185, United States of America*
grossmann@cmu.edu

Abstract

This manuscript presents the recent advances in Mixed-Integer Nonlinear Programming (MINLP) and Generalized Disjunctive Programming (GDP) with a particular scope for superstructure optimization within Process Systems Engineering (PSE). We present an environment of open-source software packages written in Python and based on the algebraic modeling language Pyomo. These packages include MindtPy, a solver for MINLP that implements decomposition algorithms for such problems, CORAMIN, a toolset for MINLP algorithms providing relaxation generators for nonlinear constraints, Pyomo.GDP, a modeling extension for Generalized Disjunctive Programming that allows users to represent their problem as a GDP natively, and GDPOpt, a collection of algorithms explicitly tailored for GDP problems. Combining these tools has allowed us to solve several problems relevant to PSE, which we have gathered in an easily installable and accessible library, GDPLib. We show two examples of these models and how the flexibility of modeling given by Pyomo.GDP allows for efficient solutions to these complex optimization problems. Finally, we show an example of integrating these tools with the framework IDAES PSE, leading to optimal process synthesis and conceptual design with advanced multi-scale PSE modeling systems.

Keywords: superstructure optimization; generalized disjunctive programming; MINLP.

1. Introduction

Process superstructure optimization is a challenging problem within Process Systems Engineering (PSE). Using a mathematical programming formulation, the problem of superstructure optimization can be written as a set of constraints to be satisfied by selecting the values of variables while optimizing an objective function. These variables can be continuous, and represent properties of a process (e.g., temperature, pressure), or discrete, representing discrete choices (e.g., selecting a piece of given equipment). The constraints include the superstructure model equations (e.g., mass and energy balances, thermodynamic equations), and the objective function is a goal to reach by selecting the decision variables (e.g., maximize profit, minimize environmental impact). It is particularly challenging to obtain the globally optimal solutions to problems whose nonlinear constraints describe a non-convex region (Kronqvist et al., 2019).

This paper covers two different modeling paradigms for such superstructure optimization problems, Mixed-Integer Nonlinear Programming (MINLP), which relies only on algebraic functions of discrete and continuous variables, and Generalized Disjunctive Programming (GDP) that considers disjunctions, logical variables, and constraints for these problems. After mentioning the different solution methods for each paradigm, we present a set of examples of process superstructure optimization problems that have been modeled and solved using each of these approaches. The modeling has been done through the open-source software Pyomo.GDP (Chen et al., 2021), and made available as part of the problem library GDPLib. Finally, we demonstrate how these modeling tools and solution methods can be applied to a more intricate model created through the advanced modeling framework IDAES PSE (Miller et al., 2018).

2. Mixed-Integer Nonlinear Programming

The optimization models obtained from process superstructures have traditionally been written in algebraic equations and variables with both continuous and discrete domains. Mathematical optimization models with these characteristics are known as Mixed-Integer Nonlinear Programs (MINLP). The solution methods for this challenging type of optimization problem usually rely on the separate treatment of the two sources of complexity of the problem, the nonlinearity of the constraints and the integer variables' discreteness. Among the best-known deterministic solution strategies for these problems, we can count the Branch-and-Bound (BB) method and decomposition methods.

Both methods rely on finding bounds to the optimal objective function value through relaxations and restrictions of the original problem. A relaxation accounts for a different optimization problem whose feasible region is larger than the one of the original problem, whose solution is an optimistic bound of the optimal solution. Among these relaxations, the usual ones are continuous relaxations, where the discreteness of the integer variables is ignored, yielding a continuous problem, and linear relaxations, where the nonlinearities of the problems are replaced by linear feasible region that encompasses the feasible region of the original problem. These relaxations are not unique, and the successful solution of these problems can strongly depend on how close the relaxation approximates the original feasible region, known as its tightness, and other factors such as its size. The restriction of the original problem usually appears when fixing the value of some of the discrete variables, leading to a continuous problem in a lower-dimensional space. In the case that the original problem's objective is minimized, the optimal solution of a relaxation yields a lower bound of the optimal objective function, while any feasible solution to the problem, usually found through a partial or total fixing of the discrete variables and an optimization on the remaining variables, leads to an upper bound of the optimal solution.

In the BB method, starting from the solution of the continuous relaxation of the original MINLP, one systematically enforces values on the discrete variables to explore increasingly smaller and restricted subproblems. The solution of specific subproblems allows the derivation of extra inequalities that can help better approximate the original problem's feasible region, improving the quality of the lower bound obtained by solving it. Although effort has been made to derive strong inequalities, or cuts, for the nonlinear case and this can be generalized for branching on continuous variables, the BB method is better known for its highly successful implementations in modern solvers when addressing Mixed-Integer Linear Programming (MILP).

The decomposition methods for MINLP usually rely on MILP relaxations, which can be efficiently solved through BB, and continuous subproblems. As with BB, the MILP relaxations can be improved iteratively using cuts derived from the solution of the continuous restrictions. Eventually, if the relaxations are rigorous and each subproblem is solved optimally, either the decomposition methods or the BB methods will find the optimal solution in a finite number of steps/iterations. These two algorithms are the main ingredients of most known MINLP solvers (Kronqvist et al., 2019).

Considering this, a solver for MINLP is usually a meta-solver, where the solution of the original problem relies on other solvers to tackle subproblems. This observation has led us to develop the open-source Mixed-Integer Nonlinear Decomposition Toolbox in Pyomo - MindtPy (Bernal et al., 2018). This solver uses the interface that the Python-based algebraic modeling language Pyomo (Hart et al., 2017) has to solvers of continuous problems and MILP solvers and provides a flexible implementation of several of the decomposition methods known in the literature, such as the Outer-Approximation method. Furthermore, it includes implementations of heuristic techniques and enhancements such as single-tree solution methods and regularization-based algorithms. Furthermore, the derivation of strong relaxations to nonlinear terms is vital to solving these problems efficiently. We have also developed the open-source software CORAMIN (Bynum et al., 2019), which generates easily refinable relaxations of a Pyomo model's nonlinear constraints. These relaxations can be integrated within MINLP algorithms.

The convergence to the optimal solution of the MINLP is guaranteed when the relaxations are valid and can be further refined after each subsequent iteration and when the continuous subproblems are guaranteed to be solved to global optimality. This is easier to achieve when assuming well-behaved nonlinearities, e.g., convexity in the nonlinear functions. In this case, the linear relaxation can be found through the 1st-order Taylor expansions of the nonlinear functions in a method known as the Outer-Approximation (OA). When the convexity assumption is not satisfied, disciplined relaxations can still be derived as implemented in CORAMIN or through generalized McCormick relaxations (Scott et al., 2011) available in the software MC++. These allow our methods to solve even non-convex MINLP problems using a Global Outer-Approximation (GOA).

3. Generalized Disjunctive Programming

A more natural framework to represent superstructure optimization problems is Generalized Disjunctive Programming (GDP), which extends the modeling capabilities of traditional mathematical programming with the incorporation of logic variables involved in propositions and disjunctions. In general, a GDP problem can be written as

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{Y},\mathbf{z}} f(\mathbf{x},\mathbf{z}) \\ &\text{s.t. } \mathbf{g}(\mathbf{x},\mathbf{z}) \leq 0;\ \Omega(\mathbf{Y}) = \text{True} \\ &\vee_{i\in D_k} \begin{bmatrix} Y_{ik} \\ \mathbf{r}_{ik}(\mathbf{x},\mathbf{z}) \leq 0 \end{bmatrix} \forall\, k \in K \\ &\mathbf{x} \in X \subseteq \mathbb{R}^{n_x};\ \mathbf{Y} \in \{\text{True}, \text{False}\}^{n_y};\ \mathbf{z} \in Z \subseteq \mathbb{Z}^{n_z} \end{aligned} \tag{1}$$

where the continuous variables are denoted by the n_x-dimensional vector $\mathbf{x}$ within a bounded set X, the discrete variables are denoted by the n_z-dimensional vector $\mathbf{z}$ within a bounded set Z, the function $f: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \to \mathbb{R}$ is the objective function, and the vector function $\mathbf{g}: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \to \mathbb{R}^{n_g}$ denotes the global constraints. Besides, the logical

structure of the problem includes $\mathbf{Y}$ as a n_y-dimensional vector of logic variables, where for each disjunct $i \in D_k$ of each disjunction $k \in K$ the set of inequalities $\mathbf{r}_{ik}: \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_{ik}}$ are enforced by the individual logic variables Y_{ik}, and $\Omega: \{\text{True}, \text{False}\}^{n_y} \rightarrow \{\text{True}, \text{False}\}$ that encodes logical relationships among the logical variables. Notice that if the disjunctions set K is empty, Eq.(1) represents an MINLP problem.

The modeling framework of GDP allows for a more intuitive representation of the problems arising from superstructure optimization. The structure of such a representation can be exploited by a tailored solution algorithm to solve these problems more efficiently. First, some reformulations can convert a GDP into MINLP, such as the Big-M and the Hull reformulations. These reformulations include all the constraints involved in the GDP and enforce or make trivially satisfiable certain constraints depending on the values of newly introduced binary variables $y_{ik} \in \{0,1\} \leftrightarrow Y_{ik} \in \{\text{True}, \text{False}\}$. The capabilities of writing a model directly as a GDP are presented as part of the open-source implementation Pyomo.GDP (Chen et al., 2021).

We present two algorithms that generalize ideas from MINLP for GDP: the Logic-based Branch-and-Bound (LBB) and the Logic-based Outer-Approximation (LOA) algorithms. Like their MINLP counterpart, these algorithms have a search strategy for the values of the discrete variables, including the logic variables $\mathbf{Y}$, but consider the logical constraint Ω, pruning it. On the other hand, by leveraging the existing structure provided by the disjunctive formulation of the GDP, some algorithms selectively remove constraints that are not involved in each combination of the logical variables while exploring that combination. This approach is beneficial given the numerical issues that can appear from evaluating nonlinear constraints on vanishing variables, i.e., "zero-flow".

Similar to the MINLP case, if the subproblems are solved to optimality, for example, through the global solvers mentioned earlier, the relaxations of the nonlinear constraints are built rigorously, using MC++ or tailored relaxations as those in CORAMIN, these algorithms can solve non-convex problems to global optimality. We call this method Global Logic-based Outer Approximation (GLOA). These algorithms are implemented in the open-source GDP solver GDPOpt (Chen et al., 2021).

4. GDPLib, the library for GDP models

Finally, we highlight the usability of our framework and solution methods by solving different process superstructure problems. These problems have been previously presented in the literature, mainly starting from an MINLP formulation. We have put together several of these examples in the Pyomo.GDP format and made it openly available in the repository https://github.com/grossmann-group/gdplib.

The library currently contains nine different examples of process or unit superstructure optimization, including a Methanol production process (Türkay & Grossmann, 1996), a Hydrodealkylation (HDA) process to produce Toluene (Kocis & Grossmann, 1989), a biofuel processing network, a heat exchanger network evaluating modular process design, a plant capacity expansion model, a synthesis gas production plant from methane, a Kaibel distillation column, and a tray distillation column design. Several of these examples include a few test cases leading to 25 GDP problems related to PSE. These examples range from $n_x \in [6,31968]$, $n_y \in [2,516]$, $n_z \in [0,5040]$, $n_g \in [30,14927]$.

We show two cases with more detail related to process superstructures. The first detailed case for the process superstructure optimization is the profit maximization of a methanol

production process (Türkay & Grossmann, 1996). Mass balances define the global constraints of the problem, and there are 4 disjunctions in this problem as seen in Fig.(1a), one associated with the feed choice, another one choosing between a high-cost and high-conversion reactor or a low-cost and low-conversion reactor, and having a single or two-stage compression for the feed and the recycle. This problem involves 285 variables, of which 8 are Boolean and the remaining continuous. The total number of configurations is 2^4 but GDPOpt using LOA requires only 2 iterations to find the optimal solution.

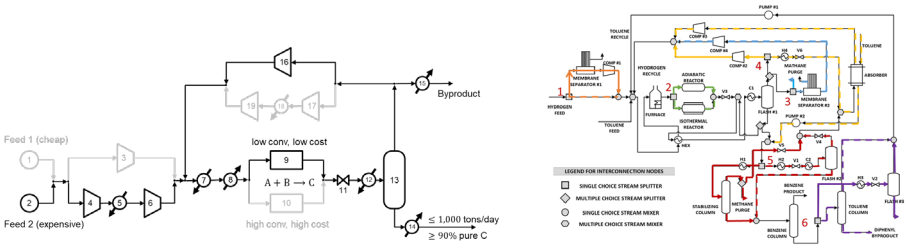

Figure 1. Flowsheet superstructure for (a) Methanol production and (b) Hydrodealkylation of Toluene. Alternatives highlighted correspond to optimal solutions.

The second example is the profit maximization of the Toluene production through the HDA process (Kocis & Grossmann, 1989). This superstructure, shown in Fig.(1b) considers 6 disjunctions: the choice to pretreat the hydrogen feed, whether to use an adiabatic or isothermal reactor, whether to purge or treat a methane stream for it to be recycled, considering the installation of an absorber or recycle a vapor stream, using a stabilizing column or a flash to remove extra methane from the process, and whether to use a distillation column to separate Toluene or a flash to obtain Diphenyl as a byproduct. There are 733 variables, with 12 being Boolean and the rest continuous, and 728 constraints, of which 12 are nonlinear. The MINLP transformation of this model is quite challenging for solvers, and both ANTIGONE and BARON fail to solve this problem to global optimality. Through LOA, GDPOpt was able to find the globally optimal solution, verified via enumeration. When using IPOPT and Gurobi as subsolvers, LOA converged to that solution in 17 iterations and 1 minute of computation in a standard desktop.

We highlight that the modeling paradigms and algorithms presented here, given their roots in Pyomo, can be used within more complicated process modeling alternatives. That is the case of the next-generation multi-scale PSE framework IDAES. This framework, by being based on Python and Pyomo, leads to supporting our implementations natively. This results in the potential use of detailed process models, including property and thermodynamic packages and a disjunctive framework. We do this through the Methanol production example, which has been reimplemented as an IDAES PSE model and is available in GDPLib. The integration with IDAES PSE allows considering rigorous thermodynamic properties, resulting in more challenging optimization problems.

5. Conclusions

This paper presents two modeling paradigms for process superstructure optimization problems: Mixed-Integer Nonlinear Programming (MINLP) and Generalized Disjunctive Programming (GDP). MINLP is the one traditionally used and for which powerful solvers have been developed. On the other hand, GDP can be not only transformed into MINLP through different reformulations, leading to a difference in solution performance but can also be solved directly by algorithms that take advantage of the disjunctive and logical

structure encoded in it. We have developed open-source software tools for such models to be implemented, as is the case for Pyomo.GDP, and solved, which is the case for MindtPy and GDPOpt, within the algebraic modeling language in Python, Pyomo.

Moreover, this paper presents process superstructure optimization problems that have been implemented using GDP. These Python implementations are available in the repository GDPLib and are freely available and installable through the package manager pip (pip install gdplib). We show two examples of flowsheet superstructure optimization, namely Methanol and Toluene production processes. Finally, we include the Methanol production process case implemented using the IDAES PSE framework. This example highlights the applicability of this paper's modeling and algorithmic ideas to an advanced process modeling framework, enabling conceptual process design through superstructure optimization integrated with rigorous property models and unit operations blocks.

We hope this library leads process designers to adopt these modeling paradigms and algorithm developers to use these examples as a testbed to improve solution methods for these optimization problems. We envision more models becoming part of GDPLib, leading to a richer resource for the process design and optimization communities.

Disclaimer: Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

Bernal, D. E., Chen, Q., Gong, F., & Grossmann, I. E. (2018). Mixed-Integer Nonlinear Decomposition Toolbox for Pyomo (MindtPy). In *Computer Aided Chemical Engineering*. https://doi.org/10.1016/B978-0-444-64241-7.50144-0

Bynum, M., Castillo, A., Laird, C., Watson, J.-P., & USDOE. (2019). *Coramin v. 0.1 Beta, Version v. 0.1*. https://doi.org/10.11578/dc.20190311.1

Chen, Q., Johnson, E. S., Bernal, D. E., Valentin, R., Kale, S., Bates, J., Siirola, J. D., & Grossmann, I. E. (2021). Pyomo. GDP: an ecosystem for logic based modeling and optimization development. *Optimization and Engineering*, 1–36. https://doi.org/10.1007/s11081-021-09601-7

Hart, W. E., Laird, C. D., Watson, J.-P., Woodruff, D. L., Hackebeil, G. A., Nicholson, B. L., Siirola, J. D., & others. (2017). *Pyomo-optimization modeling in Python* (Vol. 67). Springer.

Kocis, G. R., & Grossmann, I. E. (1989). Computational experience with dicopt solving MINLP problems in process systems engineering. *Computers and Chemical Engineering*, *13*(3), 307–315. https://doi.org/10.1016/0098-1354(89)85008-2

Kronqvist, J., Bernal, D. E., Lundell, A., & Grossmann, I. E. (2019). A review and comparison of solvers for convex MINLP. *Optimization and Engineering*, 1–59. https://doi.org/10.1007/s11081-018-9411-8

Miller, D. C., Siirola, J. D., Agarwal, D., Burgard, A. P., Lee, A., Eslick, J. C., Nicholson, B., Laird, C., Biegler, L. T., Bhattacharyya, D., Sahinidis, N. V., Grossmann, I. E., Gounaris, C. E., & Gunter, D. (2018). Next Generation Multi-Scale Process Systems Engineering Framework. In *Computer Aided Chemical Engineering*. https://doi.org/10.1016/B978-0-444-64241-7.50363-3

Scott, J. K., Stuber, M. D., & Barton, P. I. (2011). Generalized McCormick relaxations. *Journal of Global Optimization*. https://doi.org/10.1007/s10898-011-9664-7

Türkay, M., & Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers & Chemical Engineering*, *20*(8), 959–978. https://doi.org/10.1016/0098-1354(95)00219-7